SOFTWARE MAINTENANCE: RESEARCH AND PRACTICE
J. Softw. Maint: Res. Pract. **10**, 151–179 (1998)

Research

Assessing Software Maintenance Tool Utilization using Task–technology Fit and Fitness-for-use Models

MARK T. DISHAW[1,*] and DIANE M. STRONG[2]

[1]*College of Business Administration, University of Wisconsin Oshkosh, Oshkosh WI 54901–8679, U.S.A.*
[2]*Department of Management, Worcester Polytechnic Institute, 100 Institute Road, Worcester MA 01609–2280, U.S.A.*

SUMMARY

Software tools to support programmers and maintainers have been touted as potential solutions to the software development and maintenance crisis. Use of these tools is predicted to increase programmer productivity while simultaneously increasing the quality of the resulting software. Unfortunately, programmer use of these tools is lower than expected. We investigate maintenance programmers' choices about using software tools for their maintenance tasks using task–technology fit and fitness-for-use models. Our results indicate that the fit between maintenance tool functionality and the needs of maintenance tasks is associated with tool use. Two related factors, however, have stronger effects. First, the results are stronger for intention to use than for actual use. Specifically, while higher fit between tool and task is highly associated with intention to use, this intention may not lead to actual use. Second, maintainers' control over their environment affects usage. The more control maintenance programmers have over their resources and opportunities to use tools, the more likely they are to choose to use them. These results should concern maintenance managers who have acquired or are acquiring tools to increase productivity and quality, but are not realizing the benefits of their technology investments. © 1998 John Wiley & Sons, Ltd.

KEY WORDS: actual software tool use; intended software tool use; task–technology fit; fitness-for-use; cognitive fit; behavioural control

1. INTRODUCTION

Enormous sums of money are expended annually by organizations in maintaining production software. The maintenance process dominates the activities of many Information Systems (IS) organizations. Up to 70% of IS software budgets are devoted to maintenance (Swanson and Beath, 1989, pp. 1–2). In mature organizations, this percentage may be much higher. Despite the levels of financial commitment represented by software mainte-

* Correspondence to: Mark T. Dishaw, College of Business Administration, University of Winsconsin Oshkosh, Oshkosh WI 54901-8679, U.S.A. E-mail: dishaw@uwosh.edu

CCC 1040–550X/98/030151–29$17.50

Received 3 August 1997
Revised 16 November 1997

nance, historically the *process* of doing maintenance received relatively little attention from IS managers, except to complain of its difficulty and expense. The attitude of many IS managers, and IS academics, had been one of ‘benign neglect’.

While maintenance is a continual source of frustration, IS managers are increasingly interested in maintenance as a potential substitute for development. Improved maintenance practices are seen by some as a solution to the maintenance problem (Glass, 1990), and could lead to longer useful lives for ageing legacy systems.

Unfortunately, there are significant barriers to improving software maintenance practices. Software maintenance is neither well understood in practice nor adequately characterized in the literature. Research and theory in this area significantly lag behind what maintenance practitioners are trying to do (Glass, 1995, pp. 129–163).

Despite these difficulties, software tool support for the maintenance process has been identified as a key to achieving maintenance productivity gains (Schneidewind, 1987). In addition to improving productivity, the use of these tools may contribute significantly to improving the quality of the software being maintained (Kim and Westin, 1988). Although software tools for maintenance offer the potential for quality and productivity increases, our understanding of how the maintenance task can be, and currently is, supported by these software engineering tools is limited. Organizations that adopt *and are able to use* these tools achieve both productivity and quality benefits (see Iivari (1996), for a brief review of this literature). The problem, however, is low utilization of these tools (Iivari, 1996). It is also possible that the utilization of tools may lead to lower productivity in some circumstances (Bruckhaus, *et al.*, 1996).

While information technology (IT) utilization studies are common in the MIS literature (Trice and Treacy, 1988; DeLone and McLean, 1992), there are few studies of software engineering tool utilization, and almost none in the area of software maintenance tool utilization. This is typical of the general lack of attention to software maintenance in the research literature. A recent survey of the major MIS, software engineering and computer science journals found that of nearly 1 000 articles, less than three per cent concerned software maintenance in a significant way, and nearly half of those appeared in the March 1987 issue of one journal (Schneberger, 1993).

Recently, models have been developed that provide robust theoretical underpinning for IT utilization studies. These models are based on the construct of *fit* between technology and tasks and provide a theoretical foundation for investigating why software maintenance tools are not used as much as management would prefer.

This study examines this question using two operationalizations of fit in a field study of software maintenance tool utilization. Specifically, we assess whether higher fit between maintenance support tools and software maintenance tasks is associated with higher tool usage. For the purposes of our study, we define maintenance as the process of changing production software. The changes may take the form of enhancements or ‘bug’ fixes. Our ultimate objective is to develop our understanding of the fit or lack thereof between software maintenance tools and the tasks they are intended to support. The insights gained in this study, and the research program of which it is an integral part, should lead to recommendations to managers and tool builders about which tool characteristics should be provided to maintenance programmers and analysts.

2. RESEARCH MODEL

2.1. Basic model of fit and utilization

A fit construct provides the theoretical foundation for examining both downstream (factors affected by fit) variables (e.g., attitude toward tools, intention to use, tool utilization and performance) and upstream (antecedent) variables (e.g., individual, process, task and technology characteristics). With these variables, we can study the impacts of high and low fit on utilization and the factors that influence fit. Such research has practical implications for the design of better software tools to support knowledge workers and better management practices in the organization and to support knowledge work.

The fundamental argument of fit models holds, *ceteris paribus*, that IT will be used and will provide benefits if the functions available to the user support the activities of the user. An IT function is considered to support an activity if it facilitates that activity or lowers the cost to the user of performing that activity (Dishaw, 1994, p. 33). The basic fit model has task–technology fit (TTF) as the independent variable and an outcome measure, e.g., utilization, as the dependent variable, as shown in Figure 1. A task–technology fit model predicts that higher degrees of 'fit' lead to expectations by users of beneficial consequences of use (Goodhue, 1992b).

While this is a simple model, fit captures technology characteristics, task characteristics, individual characteristics and their matching. A more elaborate TTF model may include variables affecting fit, especially technology, task and individual characteristics. For our research, the basic model is sufficient for testing whether increasing fit between maintenance support tools and maintenance tasks will increase the utilization of the technology. The hypothesis tested is:

Hypothesis 1: Higher task–technology fit is associated with higher use of tools.

The MIS literature includes at least three conceptualizations of fit. The fit measures used in this study are formed from two of these conceptualizations. Furthermore, utilization has been conceptualized and measured as intention to use and actual use. The differences between intention and actual use may explain why software maintenance tool use is lower than managers expect. We further explore the conceptualizations of fit and utilization before discussing our field study of practising maintainers and their use of maintenance tools.

Figure 1. The basic fit model has two main parts

2.2. Three conceptualizations of IT fit

2.2.1. Task–technology fit

Three conceptualizations of fit have been used in the IT literature: task–technology fit, cognitive fit and fitness for use. The theoretical foundation for these fit conceptualizations is discussed below, with examples of their application in the literature. Our study of the fit of software maintenance tools primarily uses task–technology fit. As explained subsequently, we aggregate a 12-variable TTF operationalization into four higher-level constructs from a fitness-for-use model to explore further which dimensions of fit affect tool utilization.

The ability of IT to support a task is expressed by a formal construct known as task–technology fit, which is a matching of the capabilities of the technology to the demands of the task. This construct was developed from an assumption that IT will be used if the functions available to the user support, or fit, the activities of the user. An IT function supports an activity if it facilitates that activity. Alternatively stated, IT must serve to lower the cost to the user of the performance of some task or action if it is to be used. Rational, experienced users choose those tools and methods that enable them to complete the task with the greatest net benefit. ITs that do not offer sufficient advantage will not be used.

Task–technology fit is defined by Dishaw and Strong (1996a) as:

'…the matching of the functional capability of available information technology with the activity demands of the task at hand'

A task, in the task–technology fit literature, is an activity to be accomplished by a knowledge worker. Tasks can be of a problem-solving nature, as in auditing (Nance, 1992, pp. 26–27) or be software maintenance (Dishaw, 1994, pp. 21–28), or be of a decision-making nature, as in Goodhue (1995). Technology, in the task–technology fit literature, includes a wide range of information technologies, such as hardware, software, data, user-support services or any combination of these (Goodhue and Thompson, 1995).

Task–technology fit, in its simplest form, employs the rational user assumption. This assumption, and the simplification that it allows, enables us to examine in relative isolation the effect of task–technology fit on utilization. Variables, such as social norms and affect, which do not make a rational user assumption, have been included in previous research work related to the task–technology fit construct. While it was expected that social norms would have some impact on the intention to use technology, Davis (1989) and Mathieson (1991) have shown its lack of effect, thus lending support to the rational user assumption.

Goodhue's development of the TTF approach addresses some of the well-known deficiencies in the user satisfaction construct (Melone, 1990; Goodhue, 1992b; Goodhue, 1995). The satisfaction construct uses affect as a determinant of behaviour and ignores other rationally-held beliefs. For example, a person may not 'like' or have positive feelings about a piece of software but may still use the software as it leads to a favourable job or task outcome. The task–technology fit construct captures a person's belief system regarding the possible outcomes of 'task–system' fit that result from information technology use. Goodhue's (1992b) definition of 'task–system' fit is:

'"Task–system" fit is the degree to which an information system or systems environment assists individuals in performing their tasks, or the fit between task requirements and the functionality of the IS environment.'

That Goodhue definition is also similar to Nance's (1992, p. 50) which was employed in his study of auditors' use of software tools:

'. . .the degree to which an available information technology is useful in supporting the unique needs of a given task.'

This definition is similar to Cooprider and Henderson's (1991) definition of fit. They propose fit as the appropriate theoretical mechanism for studying tool support for software prototyping processes. They note that fit, which they call technology–process fit, occurs when:

'. . .supporting technologies are properly matched with the fundamental processes of the system prototyper.'

As was summarized in Figure 1, the simplest task–technology fit model consists of two constructs, fit and utilization, with fit as the independent variable and utilization as the dependent variable. Versions of the task–technology fit model include downstream and upstream variables. On the upstream side, TTF models may also include factors that affect fit, such as technology, task and individual characteristics. For example, task and technology characteristics typically are assumed to directly affect fit. Individual characteristics, such as experience, moderate these relationships. On the downstream side, individual, team or organization performance impacts may be included. Early versions of the TTF model posited a direct relationship between TTF and performance (Goodhue, 1992b). While a link between TTF and performance certainly exists, it is better understood if utilization is admitted into the causal chain, as in Dishaw (1994, pp. 35–36) and Goodhue and Thompson (1995).

Goodhue (1992b, 1995) operationalized fit in the context of accessing quantitative data for use in managerial decision making. That is, the task is managerial decision making and the technology is quantitative data stored in a computer system. Goodhue's operationalization of fit between this task and technology contains the 12 dimensions shown in Figure 2 (Goodhue, 1992b, 1995). Complete definitions of these 12 variables are in Goodhue (1992b, 1995).

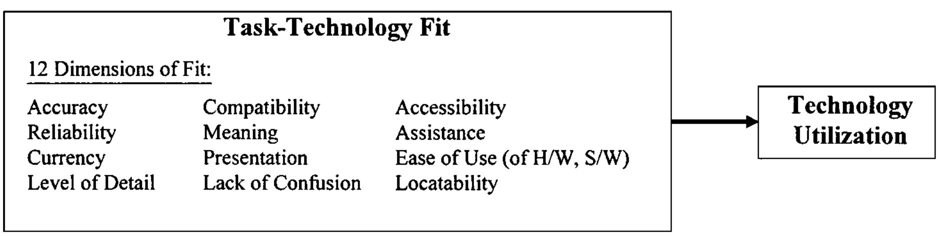

Figure 2. Goodhue's operationalization of the basic fit model introduces 12 dimensions of fit

This operationalization of fit, with some minor changes, was applied successfully across many ITs in the systems environment of end users across many functional areas of two companies (Goodhue and Thompson, 1995). Goodhue and Thompson (1995) argue that this operationalization applies to any IT, including hardware, software and data, as well as to user-support services or any combination of these technologies. This argument is supported by examining the 12 dimensions. For example, accessibility could apply equally well to hardware, software, data or services, perhaps with minor changes to questionnaire wording.

We apply this operationalization of fit to study the fit between software maintenance tools and the task of software maintenance. Goodhue's operationalization with its 12 dimensions of fit is sufficiently general to apply to a broad set of technologies and tasks, and it has been demonstrated to apply to many different information technologies, tasks and users (Goodhue and Thompson, 1995). The tools accessed and used by maintainers play a similar role to data accessed and used by managers. Thus, the dimensions of fit for data access and use are likely to be similar to those for software maintenance tool access and use, at least at the general level of the dimensions developed by Goodhue.

2.2.2. Cognitive fit

A key to understanding the nature of TTF may lie in the closely related, and more narrow construct, cognitive fit. Consider the definition of cognitive fit (Vessey and Galletta, 1991):

> 'Cognitive fit is a cost–benefit characteristic that suggests that, for most effective and efficient problem solving to occur, the problem representation and any tools or aids should all support the strategies (methods or processes) required to perform that task.'

This construct is closely related to the general notion of task–technology fit, but is not identical to it. The primary difference is that cognitive fit assumes that the support of the task requires support of the creation and manipulation of problem representations. The general TTF model makes no such restrictive assumption and has the ability to capture the effects of experience, motivation and social norms.

Cognitive fit represents an appropriate alternative model to TTF if the investigator has chosen a laboratory or experimental setting where control over task and technology, as well as moderating influences, can be achieved. It is especially useful in attempts to understand how a very specific task can be supported at a cognitive level by a technology feature or function. The more general TTF models may be more appropriate for studies in a field setting where the investigator is concerned primarily with external validity or in settings where control over the above variables is tenuous. For settings in which economic or social context variables are of overriding importance, other conceptualizations of fit may need to be used.

2.2.3. Fitness for use

The quality literature provides a third way to conceptualize fit, called fitness-for-use (FFU), which is the general definition for quality (Deming, 1986, pp. 167–182; Juran,

1989, p. 15; Figenbaum, 1991, pp. 7–9). A product, service or technology is of high quality if it is fit for use by a person performing a task. This conceptualization of quality is similar to the concept of task–technology fit. The definition of quality states that quality cannot be evaluated only from the characteristics of the product or technology; the needs of the task must also be considered.

Following Goodhue's operationalization of TTF with data as the technology, we consider a fitness-for-use model designed for data. The importance of computerized data for many organizational tasks has led to recent studies of the quality of these data (Ballou and Pazer, 1995; Strong and Miller, 1995; Wang and Strong, 1996; Strong, Lee and Wang, 1997a, 1997b). Data quality research studies have concluded that the evaluation of the quality of data must consider the context of the tasks in which the data are used (Wang and Strong, 1996; Strong, Lee and Wang, 1997b). High-quality data are defined by Wang and Strong (1996) as:

'…data that are fit for use by data consumers.'

Using this fitness-for-use definition, data quality is operationalizing in terms of 15 dimensions, which can be grouped into four higher-level categories (Wang and Strong, 1996), as shown in Table 1.

The four categories are intrinsic, contextual, representational and accessibility. Intrinsic denotes that data have quality in their own right. Accuracy is one of the dimensions underlying the intrinsic category. Contextual highlights the requirement that data quality must be considered within the context of the task at hand—that is, data must be relevant, timely, complete and appropriate in terms of amount to add value. Representational and accessibility emphasize the importance of the information systems that store and provide access to the data—that is, the systems must present data in such a way that they are interpretable, easy to understand, and represented concisely and consistently, and the systems must be accessible but secure (Wang and Strong, 1996). The basic fit model using this fitness-for-use operationalization is shown in Figure 3.

2.2.4. Integrating the three conceptualizations of fit

Are these three conceptualizations of fit three different constructs, or are they three different operationalizations of one fit construct? Our purpose here is not to answer this question fully, but to develop a basis for measuring the fit between maintenance tools

Table 1. Data quality categories and dimensions

DQ category	Data quality dimensions
Intrinsic	Accuracy, believability, objectivity, reputation of the source
Contextual	Value-added, relevancy, timeliness, completeness, appropriate amount of data
Representational	Interpretability, ease of understanding, representational consistency, concise representation
Accessibility	Accessibility, security

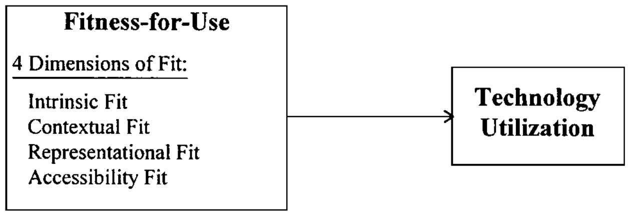

Figure 3. Wang and Strong's operationalization of the basic fit model introduces four dimensions of fit

and maintenance tasks. Cognitive fit is a somewhat different construct from the other two. It covers aspects of representational fit since it focuses on how well tools fit with problem representations, but cognitive fit is operationalized at the cognitive level. As such, it is not applicable to our field study and will not be discussed further.

We argue that the TTF and the FFU conceptualizations discussed above are differently derived operationalizations of a single fit construct. The TTF dimensions were developed by examining the literature for those characteristics of computerized data that make them useful for managerial decision making (Goodhue, 1995). The FFU dimensions were developed by asking data consumers to evaluate the attributes of data that make data most fit for use (Wang and Strong, 1996). An examination of the 12 TTF dimensions and the 15 FFU dimensions provides face validity for the similarity of the two operationalizations of fit.

To check further the similarity between these two operationalizations, we map the 12 TTF dimensions into the four FFU categories using the 15 underlying FFU dimensions as guidelines for the appropriateness of the mapping. Using the definitions of the TTF dimensions (Goodhue, 1995; Goodhue and Thompson, 1995), the 12 TTF dimensions map directly into the four FFU categories, as shown in Figure 4. Specifically, the TTF dimension of accuracy maps directly into FFU intrinsic properties, which include the FFU dimension of accuracy. For the contextual FFU category, the TTF dimension of currency and the FFU dimension of timeliness have similar definitions, as do level of detail and amount of data. The TTF dimensions of compatibility, meaning, presentation and lack of confusion are aspects of how the system represents data and are similar to the correspond-

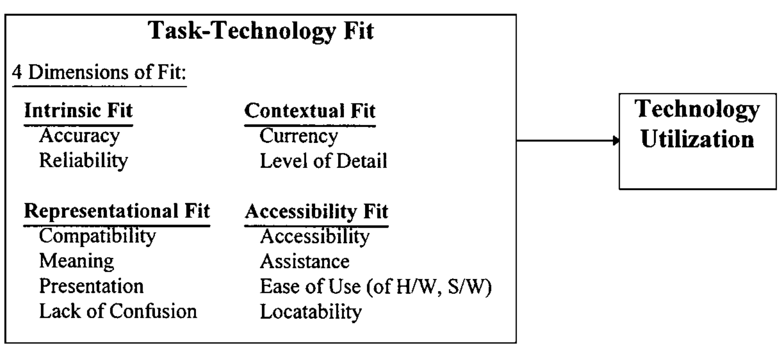

Figure 4. Dimension integration and operationalization introduce four categories of fit into the basic fit model

ing FFU dimensions of interpretability, understandability, and concise and consistent representation. The TTF dimensions of accessibility, assistance, ease of use of the hardware and software, and locatability are all aspects of the ability to find and access the data and thus map directly into the FFU accessibility category.

By examining their definitions, the mapping between the TTF dimensions and FFU categories are obvious for 11 of the 12 dimensions. For reliability, there are two possible categories. Reliability could be interpreted as an intrinsic property, similar in nature to accuracy and to the FFU dimension of reputation of the source. On the other hand, reliability could refer to the reliability of the system that stores the data, rather than to the reliability of the data itself. We classified reliability with accuracy as in the FFU conceptualization of fit.

Overall, the TTF dimensions map directly into the FFU categories. The TTF dimensions provide good coverage for the system aspects of fit covered by representational and accessibility fit. These categories cover how well the system represents and provides access to data. Goodhue's TTF dimensions do not cover intrinsic and contextual fit as well as the FFU dimensions do.

This mapping is useful for two reasons. Conceptually, it provides further understanding of the nature of the 12 TTF dimensions and what aspects of data and systems they cover. Statistically, the 12 TTF dimensions are related, and some of them are likely to be highly correlated. This presents a problem in regression, which is the typical way TTF models are evaluated. 12 correlated independent variables are likely to produce a weak regression. This is the result obtained by Goodhue and Thompson (1995, p. 229), who reported three significant betas in their regression and an adjusted coefficient of multiple determination, R^2, of 0·02. Using our mapping, we can test the hypothesis with four aggregate variables, in addition to these 12. This also increases degrees of freedom and reduces the amount of data needed to test the model, which is important for our path analysis.

2.3. Another independent variable: behavioural control

Fit, however, is not the only determinant of utilization. For maintenance tools, there is some indication from the literature that utilization is not as high as it 'should' be, given the characteristics of the tools and their potential for improving maintainer productivity and the quality of the resulting software (Iivari, 1996). While many variables other than fit could affect the amount and frequency of tool use by programmers, behavioural control is likely to be a key variable. Behavioural control by managers and team members is a key determinant of the performance of programmer teams (Henderson and Lee, 1992). Henderson and Lee (1992) define behaviour control by individual team members as:

'. . .the extent to which an individual exercises freedom or autonomy to determine both what actions are required and how to execute these activities.'

Even when maintenance tool use is voluntary (i.e., not required), there may be other ways in which maintainers feel that they do not have sufficient freedom to execute their task.

In the theory of planned behaviour (TPB), behavioural control is one predictor of outcomes (actions and behaviours). Mathieson (1991) defines behaviour control in TPB as:

'. . .an individual's perception of the presence or absence of requisite resources and opportunities necessary to perform the behaviour.'

This conceptualization of behavioural control captures barriers to system use, including access to terminals and lines (physical resources), lack of ability or experience needed to use the system and lack of time to use or learn the system (Mathieson, 1991). For example, lack of experience with CASE tools has been suggested as one explanation for the lower than expected use of these tools in organizations (Thompson, Higgins and Howell, 1994; Subramanian and Zarnich, 1996). Therefore, we augment the basic fit model with behaviour control, as shown in Figure 5.

This model says not only that technology (maintenance support tools) must fit the needs of the maintenance task, but also maintainers must have the freedom, resources, opportunities and skills to use the tool when needed. The hypothesis tested is:

Hypothesis 2: Behavioural control, in addition to TTF, is associated with use of tools.

2.4. The dependent variable: intention to use versus actual utilization

The dependent variable in the models of fit is either technology utilization, individual performance or both. This study focuses on tool utilization, the performance antecedent, as the dependent variable. Utilization is commonly employed as the dependent variable in IT success studies (DeLone and McLean, 1992). Furthermore, tool utilization is the immediate concern of MIS managers for maintenance CASE tools.

The theory of reasoned action (TRA) and the theory of planned behaviour (TPB) in the behavioural psychology literature (Ajzen, 1985; Mathieson, 1991) have served as the basis for research on information technology acceptance and use. Both TRA and TPB focus on intention to perform a behaviour as their outcome variable. Intention captures the extent to which people plan or intend to perform the behaviour under investigation. When these models are applied to IT, the behaviour is use of the technology. Thus, 'intention to perform a behaviour' becomes 'intention to use the tool'. The technology acceptance model (TAM), which was developed from these theories for studying IT usage, employs intention to use rather than actual tool utilization as its dependent variable (Davis, 1989).

The distinction between these two dependent variables, intention to use and tool utilization, is potentially important in this study. While intention to perform a behaviour

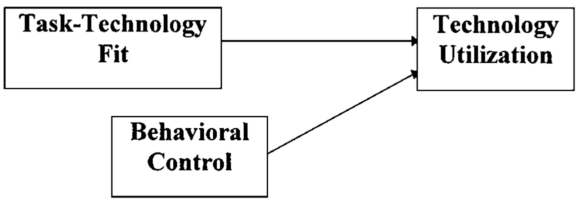

Figure 5. Adding behavioural control to the basic fit model adds an effect on technology utilization

has been shown to be an excellent predictor of the actual performance of that behaviour (Bagozzi, 1982), few IT studies collect both measures. The problems observed in organizations of lower than expected utilization might be explained by differences between intention to use and actual use. That is, there may be reasons that maintainers do not actually use tools that they would like to use. Furthermore, TAM, which employs intention to use, consistently provides good explanation of its dependent variable (Davis, 1989; Davis, Bagozzi and Warshaw, 1989). In contrast, TTF, which employs tool utilization, provides low explanatory power (Goodhue and Thompson, 1995). There could be many reasons for this observation, including that TTF is a relatively new model with fewer empirical studies. An obvious explanation is the difference in the dependent variables. Intention to use could be much easier to explain than actual tool utilization. Furthermore, intention to use is closer than actual utilization, in the causal chain, to the independent variables. For all these reasons, we test Hypotheses 1 and 2 with both of these dependent variables. The following hypothesis states our expected results:

Hypothesis 3: The tests of Hypotheses 1 and 2 will provide stronger results for intention to use as the dependent variable than for technology utilization.

Note that utilization and intention to use are appropriate dependent variables only when use is voluntary; otherwise, the most appropriate dependent variable is performance (Goodhue, 1995). Before selecting utilization and intention to use as our dependent variables, we verified that software maintenance tool use is voluntary in our subject organizations. Maintainers are not required to use these tools to complete their tasks.

2.5. TTF path model

In addition to testing Hypotheses 1 and 2, we also test the path model shown in Figure 6. This tests the path from TTF to intention to use to utilization. It helps us further examine the differences between intention to use and actual use, and how behavioural control affects these outcomes.

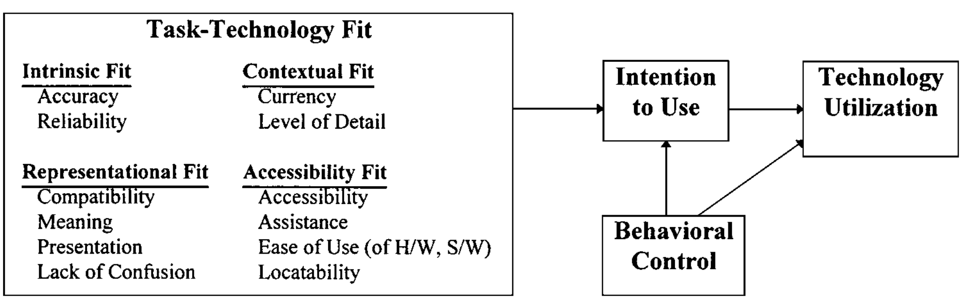

Figure 6. The TTF path model includes intention to use

3. RESEARCH METHOD

3.1. Data collection

This research was conceived and implemented as a field study employing survey methods. The subjects of the study were working professional programmers who completed maintenance projects in the normal course of their duties. This type of research design did not allow us to control the task or methods used, as in a controlled experiment (Conte, Dunsmore and Shen, 1986, p. 117). The use of a 'real world' context, however, does assist us in attaining a higher degree of external validity (Kidder and Judd, 1986, pp. 170–171).

Data for this study were collected in three organizations, an aerospace firm, an insurance company and a financial services firm. Each organization had a large number of programming groups responsible for maintaining administrative systems. Each organization had a similar systems environment for supporting maintenance both for applications and for information technologies. The systems on which maintenance was done were predominately written in COBOL. Of the projects included in this study, 92% of the projects were written in COBOL. The remainder were in a scripting language (i.e., REXX) or were of a 'mixed' type.

We collected data from maintainers on 74 individually-performed maintenance projects. The first organization contributed 39 projects, the second 27 projects and the third eight projects. Each organization is represented by at least two maintenance groups. Projects included in the study were completed by a single programmer. The length of the projects varied from less than one day to several weeks. The average maintenance project effort was 6·7 days (54 person-hours).

There were a number of different software tools available to the programmers in this study. All of the maintenance tool sets included the basic text editors such as ISPF, language compiler facilities and associated diagnostic tools common to IBM S/390 MVS installations. Additional tools were provided in each organization that included debugging, tracing and abend analysis tools. One organization also used tools that had been developed internally. All three used commercially available tools, such as the ViaSoft tool suite or the Micro Focus COBOL workbench and its related tools. All programmers had previous experience with their software tool environment. No changes or manipulations of the software tool environment were made during data collection.

Maintainers completed a questionnaire at the beginning and end of their project for all projects undertaken during a data collection period of approximately six weeks. The start and stop times were arranged individually with the managers of each participating group. The data collection period for each group did not cover the same six calendar weeks. The beginning project questionnaire included the Goodhue fit dimensions, behavioural control and intention to use. The ending project questionnaire included questions on actual use of software maintenance tools.

The maintenance projects included in this study were drawn from the existing maintenance backlog and were not selected in advance by either the maintainers or researchers. Programmers were asked to provide data on all maintenance projects that started and ended within the data collection period. This was done to mitigate selection bias problems.

We had no prior knowledge of the maintenance tasks to be performed, nor did the maintainers, as they performed their duties normally as projects were assigned to them.

Since some programmers reported on multiple projects, we performed statistical tests to determine whether the project data violated the independence assumptions of regression analyses. We performed one–way anova analyses to test for programmer (non–independence) effects on tool utilization. A *post–hoc* test using the Scheffe method indicated no significant differences at the 0·05 significance level. We also performed similar tests for maintenance group and company effects. In both cases, there were no significant differences. Other tests for bias in the data also revealed no significant problems.

3.2. Research instruments/questionnaires

The items on the maintenance questionnaire were developed from Goodhue's original questions for measuring the 12 dimensions of fit (Goodhue, 1992b, 1995). These items have established reliability, and convergent and discriminant validity. An extensive discussion of the techniques and methods used in the creation of the items and scales is contained in Goodhue (1988, pp. 32–43, 65–110). We made minor modifications in wording to reflect more closely our context of the maintenance tasks and the software tools that were available to maintainers. Our modified items and scales have statistical properties similar to the originals. Cronbach alphas were in the acceptable or better range for all but one of the 12 constructs. The items for measuring intention to use and behavioural control are minor modifications of the TAM items for these variables (Davis, 1989). The questionnaire items for intention to use, behavioural control and each of the 12 fit dimensions are listed in Appendix A together with correlation and descriptive statistics tables in Appendix B. All items are measured on a seven-point Likert scale. Note that since all of the maintenance groups had access to a portfolio of software maintenance tools, the questions were phrased to address the portfolio as a whole rather than a particular tool. Since we could not control either the type of task being performed, or the type of tools that would be used from project to project, we elected to treat the tool sets as portfolios of tools.

Tool utilization means the amount, extent or frequency of tool use. The tool utilization construct is operationalized with questions that assess the amount of time using the tool measured on a seven-point Likert scale (Guinan, Hopkins and Cooprider, 1992; Guinan, Cooprider and Sawyer, 1997). These questions were asked for each tool used by programmers. The dependent variable, tool use, is computed as the mean of the use levels of each maintenance tool reported used by the programmer for their particular project.

The two questionnaires were pretested with practising maintainers. The pretests resulted in minor wording changes to some questions. A pilot test was done with one maintenance group that was similar to the other groups that participated in the study. No problems were uncovered with the questionnaire or the procedures for collecting the questionnaire data. The pilot test maintenance group did not participate further in the main study and the pilot test data are not included in the findings reported below.

3.3. Analysis

Scores for each variable were computed as the mean of the associated questionnaire items. For the TTF operationalization of fit, each of the 12 fit dimensions has two or three associated items. For the FFU operationalization of fit, each of the four FFU categories is formed as the mean of the items associated with the corresponding TTF dimensions. Cronbach alphas for all constructs and correlations among the constructs are given in Tables 10 and 11 in Appendix B.

The analyses to produce our results were conducted in three stages:

(1) Hypothesis 1, the effect of fit on utilization, was tested using regression with fit dimensions as independent variables and utilization as the dependent variable. The task is software maintenance and the technology is CASE tools that support software maintenance activities. This hypothesis was tested for two operationalizations of fit. We first operationalized TTF using the method described by Goodhue (1992b; 1995), which measures fit through the 12 variables listed in Figure 2. For our second operationalization of fit, we consolidated Goodhue's 12 dimensions into Wang and Strong's four higher-level variables, as shown in Figure 4. Hypothesis 1 also was tested for two dependent variables. Thus, testing Hypothesis 1 involved the four regressions shown in Table 2, which use one of the two operationalizations of fit (12 variable TTF or four variable FFU) and one of the two outcome variables, utilization or intention to use. The regression using the TTF dimensions and utilization as the outcome replicated Goodhue and Thompson (1995). The regression using the consolidation of fit into four variables, as shown in Figure 4, provided a smaller and easier to interpret model that was the basis for testing Hypothesis 2.

(2) Hypothesis 2, which tests whether adding behavioural control helps to explain utilization, also was tested using regression. For this hypothesis, we focused on the four-variable FFU operationalization of fit, which was tested with both dependent variables.

(3) Finally, we tested a path model of the complete research model, as shown in Figure 6. For this, we also used the four-variable FFU operationalization of fit.

The regressions for testing Hypotheses 1 and 2 were run using SPSS for Windows. The path model was run using Amos for Windows (Bollen, 1989; Arbuckle, 1997). A

Table 2. Four regression runs to test Hypothesis 1

Fit operationalization	Dependent variable: tool utilization	Dependent variable: intention to use
12-variable	Figure 2 (replicates Goodhue and Thompson (1995))	Figure 2 with change in dependent variable
four-variable	Figure 4	Figure 4 with change in dependent variable

Table 3. Hypothesis 1, regression of 12-variable fit with tool utilization as the dependent variable

Independent variables	β	t	Significance of t	Adjusted R^2	F	Significance of F
Accuracy	0·365	1·07		0·39	2·78	0·018
Reliability	–0·205	–1·30				
Currency	–0·108	–0·49				
Level of detail	–0·156	–0·62				
Compatibility	–0·021	–0·06				
Meaning	0·489	2·20	0·039			
Presentation	–0·277	–1·05				
Lack of confusion	0·207	0·86				
Accessibility	–0·669	–2·75	0·012			
Assistance	0·194	0·65				
Ease of use (of H/W, S/W)	0·303	1·31				
Locatability	–0·351	–1·29				

comparison of all results for the two dependent variables provided the basis for testing Hypothesis 3.

4. RESULTS

4.1. Hypothesis 1: the effect of fit

Results from our four regression tests of the first hypothesis are shown in Tables 3 to 6. In the first regression, which replicates Goodhue and Thompson (1995), the dependent variable is tool use and the independent variables are the 12 dimensions of fit. The adjusted R^2 is 0·39 and the regression is significant. The signs of the beta coefficients for the variables in the model, however, are mixed. A positive sign implies that higher tool use is associated with a higher degree of the fit variable. A negative sign implies the opposite.

These results essentially replicate those of Goodhue and Thompson (1995, p. 229), who reported an adjusted R^2 of 0·02, significant at 0·05 for a sample of over 600 end users. They also reported a mix of positive and negative beta coefficients; two negative and one

Table 4. Hypothesis 1, regression of four-variable fit with tool utilization as the dependent variable

Independent variables	β	t	Significance of t	Adjusted R^2	F	Significance of F
Intrinsic fit	0·053	0·35		0·08	2·27	0·073
Contextual fit	0·048	0·15				
Representational fit	0·094	0·36				
Accessibility fit	–0·482	–2·03	0·047			

Table 5. Hypothesis 1, regression of 12-variable fit with intention to use as the dependent variable

Independent variables	β	t	Significance of t	Adjusted R^2	F	Significance of F
Accuracy	0·230	0·75		0·24	2·14	0·043
Reliability	0·333	1·98	0·056			
Currency	–0·119	–0·48				
Level of detail	0·057	0·25				
Compatibility	–0·278	–1·29				
Meaning	0·195	0·83				
Presentation	–0·173	–0·57				
Lack of confusion	0·076	0·38				
Accessibility	–0·300	–1·00				
Assistance	0·480	1·35				
Ease of use (of H/W, S/W)	0·165	0·76				
Locatability	–0·207	–0·79				

Table 6. Hypothesis 1, regression of four-variable fit with intention to use as the dependent variable

Independent variables	β	t	Significance of t	Adjusted R^2	F	Significance of F
Intrinsic fit	0·753	6·42	0·000	0·35	10·70	0·000
Contextual fit	–0·339	–1·40				
Representational fit	–0·213	–1·05				
Accessibility fit	0·243	1·33				

positive beta were significant. In ours, one negative and one positive beta are significant. They posited that the negative betas are the result of feedback—i.e., utilization affects the perception of some TTF factors. The result might also be explained by other factors, such as poor ease of use (Dishaw, 1994, pp. 126–127). For example, maintainers may not use easily accessible tools because they are difficult to use.

The 12 correlated independent variables in the above regression make the results difficult to interpret. A potential solution to this problem is to consolidate the variables, provided we have a sound prior justification for doing so. We used the four aggregate FFU categories (Wang and Strong, 1996) to consolidate the variables, as described in Section 2.

The regression of the consolidated fit variables is shown in Table 4. The dependent variable is tool use and the independent variables are the four FFU categories. The adjusted R^2 is lower (0·08), but still higher than that reported in previous work, and the significance of the regression F is 0·073. The signs of the beta coefficients for the variables in the model are still mixed. Thus, while we can conclude that fit is associated with utilization, higher fit does not always lead to higher utilization.

While these results provide some support for Hypothesis 1 and they replicate previous results (Goodhue and Thompson, 1995), they do not substantially increase our understanding of maintenance tool utilization. Only one or two dimensions of fit have significant

betas. Because of the correlated independent variables, however, we cannot conclude that the non-significant variables are not important, only that they do not provide additional explanatory power. We investigate further these results so that we can better understand why maintenance tool utilization in organizations is lower than it 'should' be.

As a first step toward better understanding of these data, we switch the outcome variable to intention to use rather than actual tool utilization. The results from regressions using the TTF dimensions and the FFU categories are shown in Tables 5 and 6.

These results show a slightly worse regression in the 12-variable case (adjusted R^2 of 0·24 rather than 0·39) and a much better regression in the four-variable case (adjusted R^2 of 0·35 rather than 0·08). The pattern of the betas is much the same for both dependent variables. The correlations between the fit variables and intention to use (see Tables 10 and 11 in Appendix B) are more positive than those between fit and utilization. As expected from the behavioural psychology literature, intention to use and actual use are significantly correlated (+0·39, significant at the 0·01 level), although the correlation is not very high. Thus, while intention to use may be slightly better as a dependent variable, we still do not have a good explanation for the low use of maintenance tools in organizations.

4.2. Hypothesis 2: the effect of behavioural control

Results from our regression tests of the second hypothesis are shown in Tables 7 and 8. In each table, regression results with behavioural control as the only independent variable are reported first, followed by the results with behavioural control and the four-variable fit operationalization as the independent variables.

In the first regression in Table 7, behavioural control is significant and the regression has a higher adjusted R^2 than for the fit variables only (see Table 4). The beta is positive, meaning that the more control maintainers can exercise to do their tasks as they see best, the higher the use of maintenance tools. Adding the fit variables to this regression produces a significant change in the regression (F of the change is 2·66, significant at 0·042). These results support Hypothesis 2, although significant fit variables are negative, as they were for the Hypothesis 1 tests.

Table 7. Hypothesis 2, regression of behavioural control and four-variable fit with tool utilization as the dependent variable

Independent variables	β	t	Significance of t	Adjusted R^2	F	Significance of F
Behavioural control	0·373	3·09	0·003	0·12	9·56	0·003
Behavioural control	0·451	3·26	0·002	0·21	4·25	0·002
Intrinsic fit	–0·241	–1·40				
Contextual fit	0·284	0·95				
Representational fit	0·126	0·52				
Accessibility fit	–0·563	–2·55	0·014			

Table 8. Hypothesis 2, regression of behavioural control and four-variable fit with intention to use as the dependent variable

Independent variables	β	t	Significance of t	Adjusted R^2	F	Significance of F
Behavioural control	0·869	14·70	0·000	0·75	216·04	0·000
Behavioural control	0·782	11·94	0·000	0·79	54·74	0·000
Intrinsic fit	0·244	3·06	0·003			
Contextual fit	0·071	0·50				
Representational fit	–0·158	–1·37				
Accessibility fit	0·101	0·97				

Switching the dependent variable to intention to use from actual use produces a highly significant regression with a high adjusted R^2. In the first regression in Table 8, behavioural control is significant and the regression has a higher adjusted R^2 than for the fit variables only (see Table 6). The beta is positive, meaning that the more control a maintainer can exercise to do their task as they see best, the higher their intention to use maintenance tools. Adding the fit variables to this regression produces a significant change in the regression (F of the change is 4·28, significant at 0·004). These results support Hypothesis 2, and the significant fit variable is positive.

4.3. Path analysis

We further explore the results from Hypotheses 1 and 2 by testing the path model. The path model was first run with all the paths as specified in Figure 6, including arcs for correlations between the independent variables (the four fit variables and behavioural control). Next, the model was rerun with the non-significant paths deleted (the dotted line paths in Figure 7). The values for path coefficients shown on the arcs in Figure 7 (0·22, 0·77 and 0·40) are from the second run, but they differ little from the coefficients from the first run. These values show the strength of the path between two variables, and are the path analysis equivalents of beta values in regression. The values on the dependent variables, intention to use (0·70) and tool utilization (0·16), are the path analysis equivalents of R^2 in regression. Overall, the path model reported in Figure 7 provides a good fit to the data, $\chi^2 = 12{\cdot}6$, df = 8, $p = 0{\cdot}13$. (Note that non-significant χ^2 represents good fit.) The model without the four non-significant paths eliminated is not quite as good ($\chi^2 = 10{\cdot}7$, df = 4, $p = 0{\cdot}03$), but still represents an acceptable path model.

This path model provides an overall view of the piecemeal results from the regressions. Fit significantly affects utilization, as expected from Hypothesis 1. Since the four fit variables are correlated, these results should not be interpreted to mean that intrinsic fit affects utilization while the others do not. We can only conclude that fit affects utilization. The path model also clearly shows the importance of behavioural control for understanding intention to use and indirectly its importance for actual tool utilization, thus supporting Hypothesis 2.

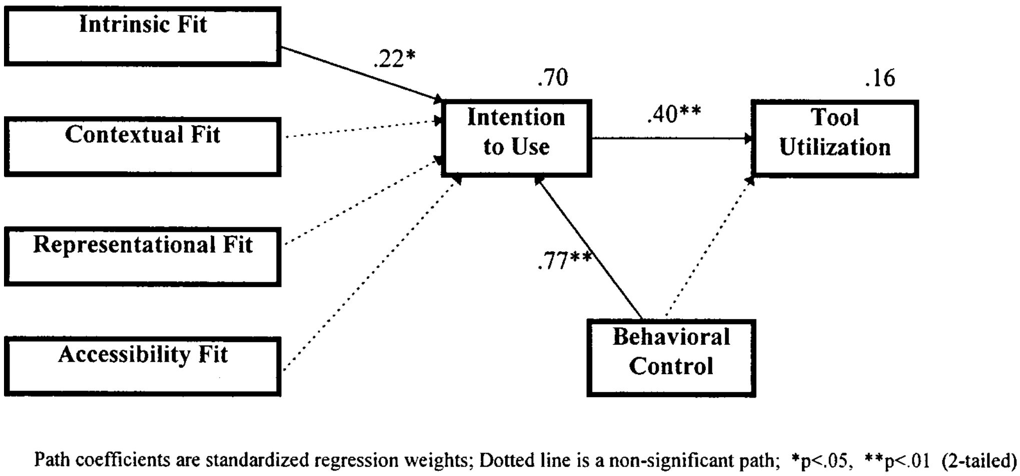

Figure 7. The TTF path model provides a close fit

4.4. Hypothesis 3: intention to use versus actual utilization

To investigate Hypothesis 3, whether intention to use has stronger results than actual utilization, the results from Tables 3 to 8 and Figure 7 are summarized in Table 9. From examining this table and looking at the path model, it is clear that the results are much stronger for intention to use than actual use as the dependent variable. All of the intention to use results are good, whereas tool utilization has some weak regressions. The first case in Table 9 is the only case for which intention to use has stronger results for actual use, but in this case both dependent variables provide good results. Thus, Hypothesis 3 is generally supported.

Table 9. Summary of results of analysis

Hypothesis	Independent variables	Results* for two dependent variables: Tool utilization	Intention to use
H1: fit → use	12-variable fit	$R^2 = 0{\cdot}39$	$R^2 = 0{\cdot}24$
	Four-variable fit	$R^2 = 0{\cdot}08$	$R^2 = 0{\cdot}35$
H2: fit and	Control	$R^2 = 0{\cdot}12$	$R^2 = 0{\cdot}75$
control → use	Control and four-variable fit	$R^2 = 0{\cdot}21$	$R^2 = 0{\cdot}79$
	Path: control and four-variable fit	$R^2 = 0{\cdot}16$	$R^2 = 0{\cdot}70$

*All coefficients of multiple determination are significant at 0·05 or better, except $R^2 = 0{\cdot}08$ which is not significant at 0·07.

5. DISCUSSION

5.1. Research implications

Overall, all three hypotheses are supported. Fit is significantly associated with tool use (Hypothesis 1), but fit along with behavioural control yields better explanatory power (Hypothesis 2). In most cases, intention to use as a dependent variable provides stronger results than tool utilization does (Hypothesis 3).

Our tests of the relationship between fit and utilization replicate and extend previous research on TTF by Goodhue (1995) and Goodhue and Thompson (1995). Our replication produces nearly identical results to that of previous work. Our extension using the data quality framework of Wang and Strong (1996) provides one possible approach for addressing the weak results between fit and utilization.

The strong effect of behavioural control provides one explanation for the lower than expected use of maintenance support tools in organizations. Programmers are more likely to use these tools when they have more opportunities, resources, knowledge and control. This effect of behavioural control is generally stronger than the effect of fit. This should concern maintenance managers. They are not providing an environment in which programmers will choose to use tools when the tools are a good match to the needs of the task. Owing to lack of knowledge, opportunities, resources or control over the situation, programmers are not using maintenance support tools as frequently as their task demands would dictate.

The strong results for intention to use, compared with actual use, have both research and practical implications. For research, these results provide an explanation for the generally better results from TAM than from TTF models. TAM employs intention to use as its primary dependent variable, whereas TTF models employ utilization or performance, which is even more distant in the causal chain. Although intention to use and actual utilization are highly related, as predicted by behavioural psychology research, the results for intention to use are much better.

5.2. Practice implications

For the practice of software maintenance, this difference in results for the two outcome variables provides another explanation for lower than expected use of maintenance tools. While programmers' intention to use these tools is a good predictor of actual use, the correlation between these two variables is only +0·4. Why is not this correlation closer to +0·8? What is interfering with the actual use of these tools when programmers intend to use them? This should concern maintenance managers. From our study, we cannot be sure what is causing these differences. The observation above that maintenance managers are not providing an environment in which programmers will choose to use these tools is further supported by the differences between the results for intention to use and actual use.

An alternative explanation for the differences between intention to use and actual use is that programmers run into unexpected difficulties when attempting to use the tools they had planned to use. For example, tool use may require converting code or documents to

the form expected by the tool or the specific task does not involve the type of problems expected. Such an alternative explanation assumes that the programmer did not accurately assess fit at the start of the project. Since the programmers in our sample are highly experienced and familiar with their tools, major unexpected tool problems are possible, but not likely. A more likely cause of inaccurate fit assessment would be poor task specifications. Maintenance tasks may involve exploratory work to determine the source of the problem. In this case, the tools the programmer thought would be relevant to the task may not be because the task is different from that expected.

Our results have practical implications for managers and senior staff of maintenance groups. In choosing maintenance support tools, managers and staff will choose tools that provide the functionality needed for the tasks maintainers perform in that organization. This is captured in our fit construct; the functionality of the tool fits the demands of the maintenance task. While some level of fit is a necessary condition for tool use, our results show two other related factors that affect actual tool use. First, the strong effect of behavioural control means that maintainers must have control over their environment and the decisions to use tools that are appropriate for the task. Second, the significant differences between intention to use and actual use means that the maintainer's plan for performing the maintenance task may change during the actual execution of that task. Lack of control over the environment or lack of knowledge about what the task will involve could cause this.

To achieve productive use of maintenance support tools, managers and staff must go beyond selecting tools with appropriate functionality. They must examine their environment to check for impediments to tool usage. For example, code may not be converted into the appropriate form for the tool, or the tool may be a good fit to the task but not to the methodology the organization has chosen for carrying out maintenance tasks. In our model, these situations are captured as low maintainer control over the maintenance environment—i.e., maintainers cannot actually use tools that are appropriate for the maintenance task.

Poorly defined maintenance tasks can also contribute to low tool usage. When tasks are poorly specified, maintainers cannot make a coherent plan for executing the task. Since tool usage often requires some set-up time, the maintainer may decide the tool is not worth using without more information about the task and whether the tool will be relevant to that task. In such an environment, maintainers may continue to use *ad hoc* methods to accomplish their task rather than available maintenance support tools. In summary, our results provide direction to managers who want to investigate why maintenance support tools have low usage and thus fail to deliver productivity and quality improvements.

5.3. Theory-driven field study

The research approach employed here to study software maintenance tool use builds on a theoretical model used to study information technology use. Using this approach, we focused on fit and behavioural control as drivers of tool use. While these are key drivers of tool use, many other factors can affect programmers' choices about using tools, e.g., technical support for the tool and tool performance. Furthermore, there are other

approaches that can be taken to understanding tool use and the impact of tools. For example, one study of the impact of software development tools used exploratory case studies of 17 projects and the insertion of a new tool into the development process to study the fit of the tool with two task variables, the size of the project and the development process used (Bruckhaus *et al.*, 1996). A variety of research approaches will contribute to increased understanding of software development and maintenance and how tools should be used to improve productivity and quality.

We designed our research as a field study rather than a laboratory experiment because our objective was a degree of high external validity. We sought to design a study whose results would be more likely to apply to other organizations than would the results obtained in a laboratory setting. The downside of a field study is the lack of control over the maintenance task being performed and the organizational environment in which they are performed. Since our research is theory driven, we are able to build on previous research with well-defined and theoretically-sound constructs. The downside of a theory-driven approach as compared with an exploratory study is that we did not set out to discover new variables that may be important drivers of maintenance support tool use. For these reasons, a variety of research approaches are needed to make progress in understanding software maintenance and software tool usage.

5.4. Additional research needed

There is clearly a need for more research on the fit between IT and knowledge worker tasks. More research on fit would help managers understand which aspects of fit lead to utilization. Such research also could lead to better tools that provide better fit to the task needs of maintenance programmers. One area of focus should be on better ways to operationalize and measure fit. Another should be on understanding the link between fit and downstream variables, such as intention to use, actual use, and resulting changes in productivity and quality. The relationship between models that focus on rational expectations, such as task–technology fit models and models that include more aspects of attitude and effect, should also be explored. More generally, other drivers of information technology use in organizations should be studied.

6. CONCLUSION

In summary, fit is an important construct for understanding the utilization of information technology when users have the freedom to choose the software and determine the extent of utilization. While other factors clearly could play a part in determining utilization, an understanding of fit is very important for tool builders in creating new information technology products.

For managers of software maintainers, understanding how additional variables such as behavioural control affect utilization also is important, because maintainers' lack of control over their environment can reduce utilization even when the fit of the technology to the task is high. Managers must not only choose maintenance support tools with high fit to the maintenance task in their organization, they must also understand and remove impediments to tool usage in their maintenance environment.

APPENDIX A. QUESTIONNAIRES

A1. Questionnaire items

This appendix classifies the research questionnaire items by category. Except as noted, participants were asked to respond using a seven-point Likert scale (1 = strongly disagree, 4 = neither, 7 = strongly agree).

A2. Task–technology fit categories

Intrinsic fit

Accuracy:

The information that I will obtain from the software maintenance tools about the software to be maintained will be accurate enough for my purposes.

There may be accuracy problems in the information I will obtain from the software maintenance tools about the software system to be maintained.

Reliability:

The software maintenance tools are subject to frequent systems problems and crashes.

I can count on the software maintenance tools to be ‘up’ and available when I need them.

Contextual fit

Currency:

I cannot get information about the system being maintained from the software maintenance tools that is current enough to meet my needs.

I need some information on the up-to-the-minute status or state of the system being maintained but cannot get it from the software maintenance tools.

The information obtained through the use of the software maintenance tools on the software being maintained will be up to date enough for my purposes.

Level of detail:

Sufficiently detailed information about the system I am about to maintain will be available from the software maintenance tools.

Information about the system that I will maintain is available from the software maintenance tools at an appropriate level of detail for my purposes.

Representational fit

Compatibility:

When it is necessary to compare information obtained from two or more different sources by the software maintenance tools about the system being maintained, there may be unexpected or difficult inconsistencies.

There will be times when supposedly equivalent information derived by the software maintenance tools from two different sources about the system being maintained will be inconsistent.

Sometimes it will be difficult or impossible to compare information derived from

two different sources by the software maintenance tools about the system being maintained because the information from the different sources may be defined differently.

Meaning:

On the reports or output produced by the software maintenance tools I will deal with, the exact meaning of information I will obtain will either be obvious, or will be easy to find out.

The exact definition of the information items I obtain from the software maintenance tools on the software to be maintained will be easy to find out.

Presentation:

The information obtained from the software maintenance tools on the system being maintained is displayed in a readable and understandable form.

The information obtained from the software maintenance tools on the system being maintained is presented in a readable and useful format.

Lack of confusion (items are reverse coded):

There are so many different software maintenance tools, each with different capabilities, which can be used to obtain information on the software to be maintained, that it is hard to understand which one to use in a given situation.

There are so many different types of software maintenance tools available, it is hard to know how to use them effectively.

Accessibility fit

Accessibility:

I can quickly get information about the system to be maintained.

It is easy to get access to information that I need on the system to be maintained.

It will be difficult to obtain the information on the system to be maintained which I need to complete this project.

Assistance:

I will get the help I need in accessing and understanding information on the system being maintained.

It will be easy to get assistance if I have trouble finding or using information on the system being maintained.

Ease of use (of H/W, S/W):

It is (was) easy to learn how to use the software maintenance tools that give me access to information on the system to be maintained.

The software maintenance tools that give me access to information on the system being maintained are convenient and easy to use.

Locatability:

It is easy to locate information about the system to be maintained, even if I have not used that information before.

It is easy to find what information is available on the system to be maintained.

A3. Tool utilization

Up to 10 tools may be specified by name by the programmer. This item appears in questionnaire two.

I used this tool ____(name)____ : (1 = a little, 4 = to some degree, 7 = a lot)

A4. Intention to use

I will use the software maintenance tools rather than manual methods to complete this project.

My intention is to use the software maintenance tools rather than manual methods in completing this project.

In completing this project, I would rather use the software maintenance tools than use manual methods alone.

A5. Behavioural control

I have much more control over the use of the software maintenance tools as compared with manual methods.

Given the resources, opportunities and knowledge it takes to use the software maintenance tools versus manual methods, it would be easier for me to choose to use the software maintenance tools.

I would be much more able to use the software maintenance tools rather than manual methods because of differences in the resources, opportunities and knowledge it takes to use each one.

APPENDIX B. DESCRIPTIVE STATISTICS

The following Tables 10 and 11 present the means, standard deviations, and correlations and their two-tailed significance for the variables used to test our hypotheses. Cronbach alphas are shown on the diagonal.

Acknowledgements

An extended abstract of an earlier version of this paper was presented at the Second Americas Conference on Information Systems (Dishaw and Strong, 1996b). We thank the participants at this conference and at the INFORMS Conference on Information Systems and Technology (Dishaw and Strong, 1996a) for helpful comments on our research. We also thank the *Journal's* anonymous reviewers for their helpful suggestions for improving this paper.

Table 10. Correlations and reliabilities for Goodhue's (1992a) operationalization of task–technology fit

	Mean	SD												
Accuracy	3·71	1·99	(0·53)											
Reliability	5·33	1·50	0·11	(0·68)										
Currency	3·64	2·03	0·50**	0·03	(0·67)									
Level of detail	3·22	1·85	0·79**	0·02	0·25	(0·68)								
Compatibility	4·29	1·32	0·25	0·19	0·42**	0·20	(0·77)							
Meaning	3·78	1·83	0·80**	–0·03	0·34**	0·73**	–0·12	(0·70)						
Presentation	3·83	2·05	0·90**	0·10	0·27*	0·74**	0·04	0·76**	(0·89)					
Lack of Confusion	4·92	1·71	–0·26*	0·32*	–0·09	–0·21	0·60**	–0·42**	–0·28*	(0·74)				
Accessibility	3·42	1·67	0·73**	–0·04	0·51**	0·72**	0·04	0·74**	0·74**	0·33*	(0·67)			
Assistance	4·00	2·14	0·71**	–0·05	0·12	0·72**	–0·28*	0·66**	0·76**	0·39**	0·75**	(0·90)		
Ease of use	3·42	1·93	0·73**	0·05	0·19	0·76**	–0·07	0·69**	0·79**	–0·26*	0·68**	0·72**	(0·87)	
Locatability	3·41	2·05	0·68**	–0·10	0·30*	0·79**	0·07	0·61**	0·68**	–0·31*	0·80**	0·80**	0·72**	(0·89)
Tool utilization	5·61	1·34	–0·26*	0·06	0·16	–0·17	0·37*	–0·22	–0·39**	0·58**	–0·29*	–0·38**	–0·25*	–0·23
Intention to use	5·01	1·74	0·15	0·56**	–0·01	0·09	–0·14	0·09	0·05	0·20	–0·02	–0·05	0·14	0·01
Behavioural control	4·61	1·70	0·05	0·58**	–0·08	–0·07	–0·18	–0·05	–0·06	0·26*	–0·17	–0·09	0·06	–0·17

Notes: *significance is 0·05 or better; **significance is 0·01 or better.

Table 11. Correlations and significance for Wang and Strong's (1996) operationalization of fitness for use

	Mean	SD							
Intrinsic fit	4·64	1·28	(0·57)						
Contextual fit	3·47	1·81	0·38**	(0·70)					
Representational fit	3·87	1·42	0·56**	0·68**	(0·68)				
Accessibility fit	3·54	1·73	0·35**	0·77**	0·77**	(0·91)			
Tool utilization	5·61	1·34	–0·03	–0·22	–0·19	–0·31*	N/A		
Intention to use	5·01	1·74	0·55**	0·16	0·10	0·04	0·39**	(0·92)	
Behavioural control	4·61	1·70	0·42**	0·04	–0·02	–0·12	0·37**	0·87**	(0·89)

Notes: N/A indicates not applicable; *significance is 0·05 or better; **significance is 0·01 or better.

References

Ajzen, I. (1985) 'From intentions to actions: a theory of planned behavior', in Kuhl, J. and Beckmann, J. (Eds), *Action Control: From Cognition to Behavior*, Springer–Verlag, New York NY, pp. 11–39.

Arbuckle, J. L. (1997) *Amos Users' Guide, Version 3.6*, Small Waters Corporation, Chicago IL, 600 pp.

Bagozzi, R. P. (1982) 'A field investigation of causal relations among cognitions, affect, intentions, and behavior', *Journal of Marketing Research,* **19**(11), 562–584.

Ballou, D. P. and Pazer, H. L. (1995) 'Designing information systems to optimize the accuracy–timeliness tradeoff', *Information Systems Research,* **6**(1), 51–72.

Bollen, K. A. (1989) *Structural Equations with Latent Variables*, John Wiley & Sons, Inc., New York NY, 514 pp.

Bruckhaus, T., Madhavji, N. H., Janssen, I. and Henshaw, J. (1996) 'The impact of tools on software productivity', *IEEE Software*, **13**(5), 29–38.

Conte, S. D., Dunsmore, H. E. and Shen, V. Y. (1986) *Software Engineering Metrics and Models,* Benjamin/Cummings, Menlo Park CA, 396 pp.

Cooprider, J. G. and Henderson, J. C. (1991) 'Technology–process fit: perspectives on achieving prototyping effectiveness', *Journal of Management Information Systems,* **7**(3), 67–87.

Davis, F. D. (1989) 'Perceived usefulness, perceived ease of use, and user acceptance of information technology', *MIS Quarterly*, **13**(3), 318–339.

Davis, F. D., Bagozzi, R. P. and Warshaw, P. R. (1989) 'User acceptance of computer technology: a comparison of two theoretical models', *Management Science,* **35**(8), 982–1003.

DeLone, W. H. and McLean, E. R. (1992) 'Information systems success: the quest for the dependent variable', *Information Systems Research,* **3**(1), 60–95.

Deming, W. E. (1986) *Out of Crisis*, MIT Press, Cambridge MA, 507 pp.

Dishaw, M. T. (1994) 'Using maintenance oriented software engineering tools to support software maintenance activities', Doctoral Dissertation, Graduate School of Management, Boston University, Boston MA, 175 pp.

Dishaw, M. T. and Strong, D. M. (1996a) 'Explaining information technology utilization with the task–technology fit construct', in Pirkul, H. and Shaw, M. J. (Eds), *Proceedings of the First INFORMS Conference on Information Systems and Technology*, The Institute for Operations Research and the Management Sciences, Providence RI, pp. 236–239.

Dishaw, M. T. and Strong, D. M. (1996b) 'Using a general task–technology fit model to assess software maintenance tool utilization', in Carey, J. M. (Ed), *Proceedings of the Americas Conference on Information Systems*, Association for Information Systems, Pittsburgh PA, pp. 66–68.

Figenbaum, A. V. (1991) *Total Quality Control*, McGraw–Hill Publishing Co., New York NY, 863 pp.
Glass, R. L. (1990) 'Software maintenance is a solution—not a problem', *Journal of Systems and Software*, **11**(2), 77–78.
Glass, R. L. (1995) *Software Creativity*, Prentice–Hall, Inc., Englewood Cliffs NJ, 268 pp.
Goodhue, D. L. (1988) 'Supporting users of corporate data: the effect of I/S policy choices', Doctoral Dissertation, Sloan School of Management, Massachusetts Institute of Technology, Cambridge MA, 203 pp.
Goodhue, D. L. (1992a) 'A new instrument for user evaluations of information systems: toward organizational assessments of MIS effectiveness', Working Paper, Information and Decision Sciences Department, University of Minnesota, Minneapolis MN, 35 pp.
Goodhue, D. L. (1992b) 'User evaluations of MIS success: what are we really measuring?', in *Proceedings of the Twenty–fifth Hawaii International Conference on System Sciences*, Volume 4, IEEE Computer Society Press, Los Alamitos CA, pp. 303–314.
Goodhue, D. L. (1995) 'Understanding user evaluations of information systems', *Management Science*, **41**(12), 1827–1844.
Goodhue, D. L. and Thompson, R. L. (1995) 'Task–technology fit and individual performance', *MIS Quarterly*, **19**(2), 213–236.
Guinan, P. J., Hopkins, G. N. and Cooprider, J. G. (1992) 'Automation of the applications development life cycle: measuring the value added', in *Proceedings of the Thirteenth International Conference on Information Systems*, ACM Press, New York NY, p. 255.
Guinan, P. J., Cooprider, J. G. and Sawyer, S. (1997) 'The effective use of automated application development tools', *IBM Systems Journal*, **36**(1), 124–139.
Henderson, J. C. and Lee, S. (1992) 'Managing I/S design teams: a control theories perspective', *Management Science*, **38**(6), 757–777.
Iivari, J. (1996) 'Why are CASE tools not used?', *Communications of the ACM*, **39**(10), 94–103.
Juran, J. M. (1989) *Juran on Leadership for Quality: An Executive Handbook*, The Free Press, New York, NY, 376 pp.
Kidder, L. H. and Judd, C. M. (1986) *Research Methods in Social Relations*, Holt Rinehart and Winston, New York NY, 563 pp.
Kim, C. and Westin, S. (1988) 'Software maintainability: perceptions of EDP professionals', *MIS Quarterly*, **12**(2), 167–185.
Mathieson, K. (1991) 'Predicting user intentions: comparing the technology acceptance model with the theory of planned behavior', *Information Systems Research*, **2**(3), 173–191.
Melone, N. P. (1990) 'A theoretical assessment of the user satisfaction construct in information systems research', *Management Science*, **36**(1), 76–91.
Nance, W. D. (1992) 'Task/technology fit and knowledge worker use of information technology: a study of auditors', Doctoral Dissertation, The Graduate School, University of Minnesota, Minneapolis MN, 216 pp.
Schneberger, S. L. (1993) 'Software maintenance: an update on issues and research', in *Proceedings of the 31st Annual Southeast Conference*, ACM Press, New York NY, pp. 226–234.
Schneidewind, N. F. (1987) 'The state of software maintenance', *IEEE Transactions on Software Engineering*, **SE-13**(3), 303–310.
Strong, D. M., Lee, Y. W. and Wang, R. Y. (1997a) '10 potholes in the road to information quality', *IEEE Computer*, **30**(8), 38–46.
Strong, D. M., Lee, Y. W. and Wang, R. Y. (1997b) 'Data quality in context', *Communications of the ACM*, **40**(5), 103–110.
Strong, D. M. and Miller, S. M. (1995) 'Exceptions and exception handling in computerized information processes', *ACM Transactions on Information Systems*, **13**(2), 206–233.
Subramanian, G. H. and Zarnich, G. E. (1996) 'An examination of some software development effort and productivity determinants in ICASE tool projects', *Journal of Management Information Systems*, **12**(4), 143–160.
Swanson, E. B. and Beath, C. M. (1989) *Maintaining Information Systems in Organizations*, John Wiley & Sons, Inc., New York NY, 255 pp.

Thompson, R. L., Higgins, C. A. and Howell, J. M. (1994) 'Influence of experience on personal computer utilization: testing a conceptual model', *Journal of Management Information Systems*, **11**(1), 167–187.
Trice, A. W. and Treacy, M. E. (1988) 'Utilization as a dependent variable in MIS research', *Data Base*, **19**(3/4), 33–41.
Vessey, I. and Galletta, D. (1991) 'Cognitive fit: an empirical study of information acquisition', *Information Systems Research*, **2**(1), 63–84.
Wang, R. Y. and Strong, D. M. (1996) 'Beyond accuracy: what data quality means to data consumers', *Journal of Management Information Systems*, **12**(4), 5–34.

Authors' biographies:

Mark T. Dishaw is an Assistant Professor of Information Systems in the College of Business Administration at the University of Wisconsin Oshkosh. He received his D. B. A. in Management Information Systems from Boston University in 1994. His research interests include software development and maintenance practices, and information system privacy. His e-mail address is: dishaw@uwosh.edu

Diane M. Strong is an Assistant Professor in the Management Department at Worcester Polytechnic Institute. She received her Ph.D. in information systems from Carnegie Mellon University. Her research centres on data and information quality, software maintenance, and software quality. Diane is a founder and an organizer of the annual Conference on Information Quality. Her e-mail address is: dstrong@wpi.edu